

(through a Submit button or similar element), a JavaScript makes the “membrane” style visible, which renders the Web page form elements (with lower z-index numbers) inaccessible. Consequently, the user is unable to click the Submit button or any other element on the page. Assuming that the “membrane” Web page division (layer) contains no text, the layer will be transparent.

[00015] These and other aspects of the present invention will be elucidated in the following detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[00016] The foregoing summary, as well as the following detailed description of the preferred embodiments, is better understood when read in conjunction with the appended drawings. For the purpose of illustrating the invention, there is shown in the drawings an embodiment that is presently preferred, in which like references numerals represent similar parts throughout the several views of the drawings, it being understood, however, that the invention is not limited to the specific methods and instrumentalities disclosed. In the drawings:

[00017] FIG. 1 illustrates an overview of an exemplary system architecture and a sequence of events during a web user's interaction with a vendor web site and vendor application software;

[00018] FIG. 2 illustrates an exemplary client workstation;

[00019] FIGS. 3A and 3B illustrate an exemplary embodiment of the present invention in HTML; and

[00020] FIG. 4 is a flow chart illustrating the operation of the source code of FIGS. 3A and 3B.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[00021] The present invention is directed to an improved method and system for preventing inadvertent entry or submissions of data by a user. In order to provide a general overview of Web-based transactions, FIG. 1 illustrates an exemplary system 100 for accessing a merchant's order entry application or financial institution's account management application (i.e., vendor software application 142) via the WWW. As shown in FIG. 1, the system 100 includes a web browser 112 running on a client workstation 110; a web server application 122 running on a web server computer system 120; a connection 114 (e.g.,

dial-up, DSL, Cable, satellite connection to an Internet Service Provider) connecting the web browser 112 and the web server application 122; an Internet/application gateway 132 running on a computer system 130, a connection 116 connecting the web server application 122 and the Internet/application gateway 132; a software application 142 running on a computer system 140; and a connection 118 connecting the Internet/application gateway 132 and the vendor software application 142. Remote computer systems 130 and 140 are typically part of a merchant or institution private corporate network 125.

[00022] In addition, while not required, the remote computer systems 130 and 140 may be the same computer system. Finally, the connections 114, 116 and 118 are communication links or communication mechanisms and may be any type of physical or logical means of connecting computer systems known in the art. This includes, but is not limited to, direct connections, Internet connection, Intranet connections, Infrared (IR), wireless and other forms of wireless connections. It is anticipated that many alternative methods and material for connecting computer systems will be readily adapted for use with the present invention.

[00023] In the exemplary architecture of FIG. 1, when a user wishes to access the software application 142 or retrieve data under the control of the software application 142, the user inputs a request from the user workstation 110 by providing input to the web browser 112. The web browser 112 communicates via the connection 114 with the web server application 122, which is running on the web server computer system 120. The web server computer system 120 and the client workstation 110 may or may not be co-located on the same physical computer hardware system. Also, the web server application 122 may be co-located with the web Browser 112.

[00024] A user at client workstation 110 performs an action that causes web browser 112 to request access to the software application 142 via the WWW by inputting data 150 to web server application 122. The input data may take the form of an HTML and/or XML page, a URL, etc. Web browser 112 then uses standard communication protocols (e.g., HTTP) to pass web browser input data 150 to web server application 122 for processing. The web server application 122 will process the request from the web browser 112 and, by examining the data stream received from web browser 112, recognize that the user wants to access the vendor software application 142. The web server application 122 will then

communicate with the Internet/application gateway 132 via using communication link 116.

[00025] The Internet/application gateway 132 acts as a translator/facilitator to allow the request from the user to be processed by the vendor software application 142. Typically, this will involve translating URLs, HTML, or other user-driven commands into a format or language that can be understood and processed by the vendor software application 142. Accordingly, the Internet/application gateway 132 is capable of communicating with the vendor software application 142 using the native interface of the vendor software application 142. In other words, the gateway 132 generates from the web server input data 150 appropriate commands that the vendor software application 142 can understand and communicates the commands to the vendor software application 142, along with sufficient information to identify the process or function that the vendor software application 142 will perform to the web client 110 that requested the process or function. Thus, the user may access the Internet/application gateway 132 via a transparent web client interface. The actual processing and communicating being done by the Internet/application gateway 132 may be completely hidden from the user of the web browser 112.

[00026] After the vendor software application 142 has performed the task requested by the user via the web browser 112, an appropriate status or data corresponding to the request is returned to the Internet/application gateway 132 via the connection 118. The Internet/application gateway 132 in turn translates the requested data or status into a form that the web server application 122 can understand and transmits the requested data or status to the web server application 122 using the connection 116. Upon receipt, the web server application 122 outputs the requested status or data into a form appropriate for the web browser 112 (output data 160). The web browser 112 can then display the appropriate output for the client workstation 110. This format would include, once again, any type of data that the web browser 112 is capable of understanding (i.e., MIME, JAVA, HTML, XML, etc.). Finally, web browser 112 provides web browser output data 160 to the user in the appropriate form (e.g., displays an output page 152 (e.g., in HTML) on workstation monitor 22, see FIG. 2).

[00027] As noted above, FIG. 1 illustrates a typical sequence of events from the user's input via the web browser 112 to the response by the vendor software application 142 back to the browser 112. This sequence may be repeated several times during a transaction